

Programação I

Trabalho "Jogo do Ouri"



Relatório sobre implementação em Python 3.4  
do Jogo do Ouri

2014/2015

Trabalho realizado pelos alunos:

- 32378 Gil Catarino
- 32427 Diogo Aguiar

## Implementação do Código

O jogo inicia-se com uma chamada à função **menu()**, que faz "imprimir" no écran o Menu, de acordo com o especificado:

.....

\*\*\*\*\* JOGO OURI \*\*\*\*\*

0 - Sair

1 - Humano vs Humano: modo texto

2 - Humano vs Humano: modo gráfico

3 - Humano vs Computador

Escolha a opção desejada:

.....

Escolha a opção desejada: 0

>>>

.....

Escolha a opção desejada: 2

Opcao nao disponivel..(em desenvolvimento)

\*\*\*\*\* JOGO OURI \*\*\*\*\*

0 - Sair

1 - Humano vs Humano: modo texto

2 - Humano vs Humano: modo gráfico

3 - Humano vs Computador

Escolha a opção desejada:

.....

Escolha a opção desejada: 1

\*\*\*\*\* JOGO OURI \*\*\*\*\*

O Ouri é um jogo de captura. O Objetivo do jogo é capturar mais sementes que o adversário. Vence o jogador que obtiver 25 (ou mais) sementes.

\*\*\*\*\*

Digitar o nome do Jogador 1 ?:

.....

A função `menu()` fica à espera que o utilizador entre com um input (opção desejada).

Caso seja chamada a opção "0 - Sair", é chamada a função de sistema `sys.exit(0)`, que faz terminar a execução do programa.

Nota: Por falta de tempo, não foi possível desenvolver a opção "2 - Humano vs Humano, modo gráfico", pelo que, se esta opção for selecionada (assim como uma opção inválida) o programa chama novamente a função **`menu()`**, e fica à espera de um novo input válido.

Ao ser selecionado o input correspondente à opção 1 ou 3, é chamada a função **`inicio()`**, que apresenta uma breve descrição do jogo, e que chama a função **`jogadores()`**, com um parâmetro "qt", que se destina a informar o programa da quantidade de jogadores humanos (1 ou 2).

Assim, se forem 2 jogadores humanos, é pedido o nome de cada um (não sendo permitidos nomes iguais), senão é pedido apenas o nome de um jogador, sendo o outro o jogador "Computador".

Nota: Não é também permitido o nome "Computador" para um jogador humano, já que esse nome é usado em algumas funções como parâmetro identificador. Para facilitar a validação dos nomes é usada a função `string.upper()`, que transforma uma string em uma string de letras maiúsculas.

Depois de inseridos e validados os nomes dos jogadores, é chamada a função **`jogador iniciar()`**, que faz a atribuição da ordem em que os jogadores jogam, utilizando a função `random`, e em seguida chama a função **`main()`**.

A função **`main()`** inicializa a lista tabuleiro com 14 posições, sendo as 6 iniciais referentes às casas do 1º jogador a jogar, as 6 seguintes referentes às casas do 2º jogador a jogar, e as duas

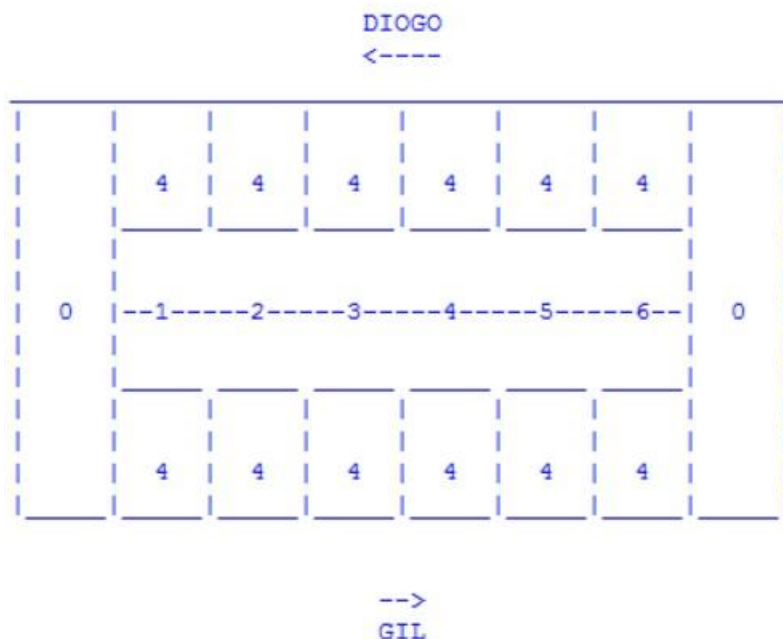
últimas referentes aos depósitos do 1º e 2º jogadores, respetivamente. As casas de ambos os jogadores são inicializadas com 4 sementes.

Em seguida o programa chama a função **inicio\_jogo()**, que apenas imprime o tabuleiro inicial, com os nomes dos jogadores(em baixo o 1º a jogar), e preenchido com as sementes iniciais (4 em cada casa), e os depósitos a zero (o depósito do 1º jogador é o que se encontra à direita (seta a indicar), retornando em seguida à função **main()**, e chamando a função **jogada\_primeiro()**, que se refere à jogada do 1º jogador, onde é pedida a casa que o jogador pretende jogar (casas indicadas na linha central do tabuleiro impresso).

.....

Digitar o nome do Jogador 1 ? : Gil

Digitar o nome do Jogador 2 ? : Diogo



Jogue por favor, DIOGO

(utilize as teclas de 1 a 6 para seleccionar a casa que deseja):

.....

Esta função, que é idêntica à função **jogada\_segundo()**, para além de validar se algum jogador já tem sementes suficientes para ganhar o jogo, faz também a validação das regras da jogada, através da função **verificar\_jogada()**, e chama a função **jogar()**.

Na função **jogar()**, é efetuada a distribuição das sementes, consoante a casa escolhida pelo jogador, e verificando as regras (exemplo: se sementes na casa escolhida forem mais do que 12, é saltada a casa de partida na distribuição das sementes).

Esta função, por sua vez, chama a função **atualiza\_deposito()**, onde são efetuadas as atualizações dos depósitos, caso se verifiquem as condições para isso.

De referir também que, através da função **verificar\_fim\_jogo()**, se analisa se estão verificadas algumas das condições que fazem o jogo terminar, e caso se verifiquem, é chamada a função **fim\_jogo()**, onde é apresentado o resultado final do jogo.

De notar que as funções **jogada\_primeiro()** e **jogada\_segundo()** são chamadas alternadamente, para cada um dos jogadores, caso não existam condições que façam obrigar um jogador a efetuar uma segunda jogada, e caso não existam condições para se considerar o jogo acabado.

Por fim, de salientar que o código inclui uma função **jogada\_computador()**, que se destina a efetuar as jogadas do computador, de acordo com o seguinte algoritmo (que não conseguimos testar e afinar na totalidade, por falta de tempo):

- Passam-se todos os elementos da lista correspondentes ao jogador "Computador", verificando-se um a um qual o nº de sementes que essa jogada permitiria capturar (caso essa jogada permitisse verificar as condições para capturar sementes), sendo esse nº de sementes a capturar guardado numa lista.
- Em seguida essa lista é percorrida à procura do índice que permitiria a maior captura, sendo então selecionada a casa correspondente.
- Caso não existam condições para decidir se uma jogada apresenta um benefício superior a qualquer outra, então a jogada é decidida aleatoriamente.
- Em seguida essa jogada do computador é validada e executada utilizando as mesmas funções que são utilizadas pelos jogadores humanos.

FIM

Évora, 3 de janeiro de 2015